Design Document for CyGrind

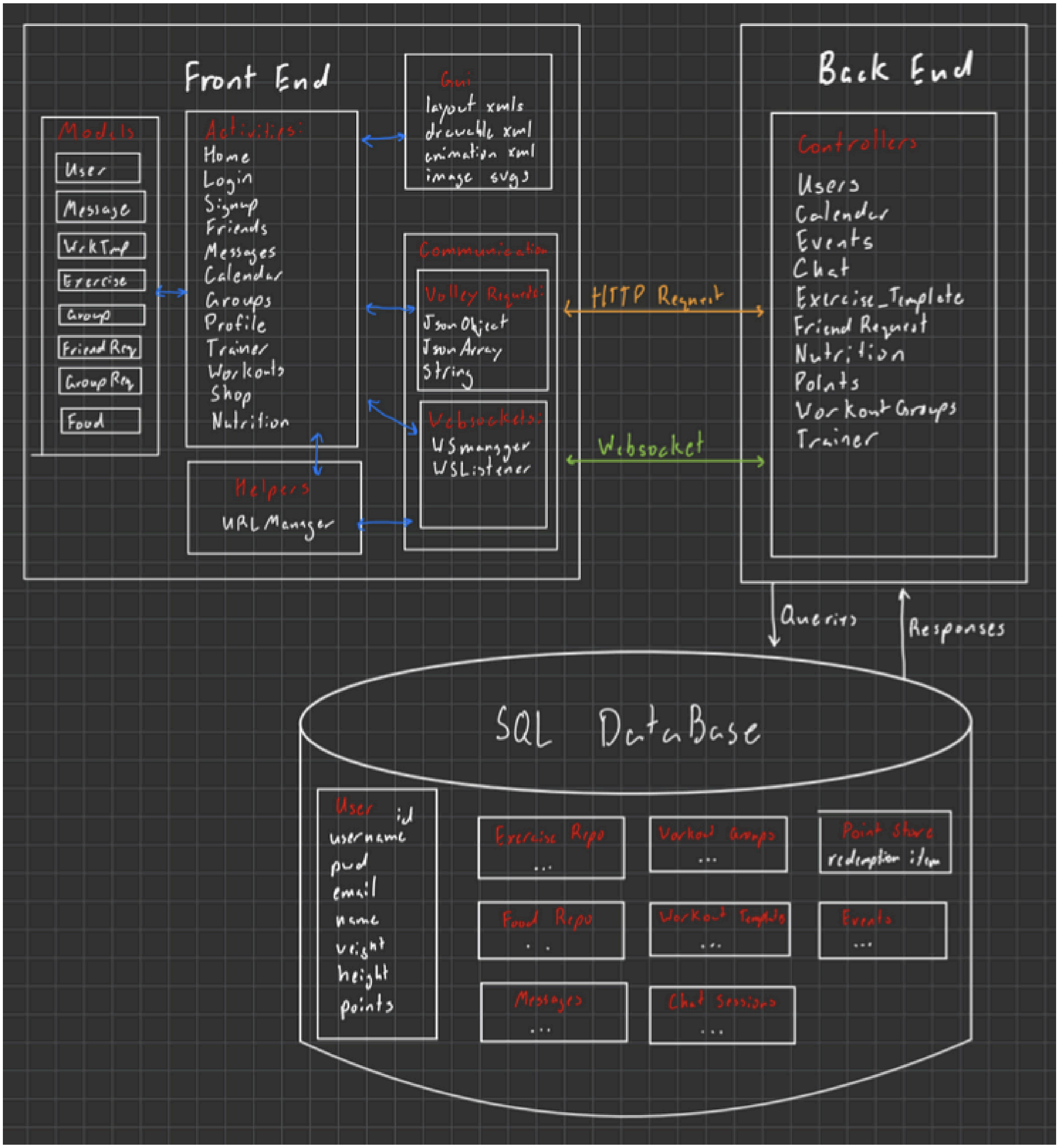
Group 4_jubair_3

Member1 Luke: 25% contribution

Member2 Joey: 25% contribution

Member3 Aidan: 25% contribution

Member4 Dawud: 25% contribution


Front End
Models
User
Message
WrkTmp
Exercise
Group
Friend Req
Group Req
Food
Activities:
Home
Login
Signup
Friends
Messages
Calendar
Groups
Profile
Trainer
Workouts
Shop
Nutrition
Gui
layout xmls
drawable xml
animation xml
image svgs
Communication
Volley Requests:
Json Object
Json Array
String
Websockets:
WSmanager
WSListener
Helpers
URL Manager
HTTP Request
Websocket
Back End
Controllers
Users
Calender
Events
Chat
Exercise_Template
Friend Request
Nutrition
Points
Workout Groups
Trainer
Queries
Responses
SQL DataBase
User
id
username
pwd
email
name
weight
height
points
Exercise Repo
...
Workout Groups
...
Point Store
redemption item
Food Repo
. .
Workout Template
...
Events
...
Messages
...
Chat Sessions
...

Frontend (implemented)

FriendPage

- The friend page contains 3 fragments (or tabs) which users can swipe between. The three tabs are as follows:
    - *Friends*: displays the current user's friends. Next to each friend are two buttons - one for messaging and one for removing the friend.
    - *Search*: displays a list of the app's users. The users can filter by typing in the search bar, and the results are filtered as the search bar changes. Users can hit the add button to generate a friend request.
    - *Requests*: This tab displays the incoming friend requests for the current user. Users have the option to accept or deny requests using the appropriate buttons.
- Any tabs can be swiped down to refresh the tab's contents with the updated backend. These mainly include the friends tab and Trainer Tabs

Workout Groups

- Like the friend page, the workout group contains many fragments, or tabs, this time 4.
    - *Search:* Users can scroll through existing workout groups, which display each group's name, type, and description. They can request to join a group by clicking the request button. Optionally, users can create their own group from this page using the "create" button.
    - *Members:* If the user is part of a group, the members will be displayed here. Each member will have a message button next to their name, so users can personally message any member. In addition, if the current user is the group's leader (meaning they created it), they will see a button next to each member to remove them.
    - *Chat:* In this tab, a chat activity is embedded in the fragment. This chat contains all the features of the regular chats, but includes every member in the group. Members are dynamically added and removed from the chat when they join or leave the group.
    - *Requests:* Only viewable by the leader, the requests to join the group are displayed here. Similar to friend requests, the leader can accept or deny requests.

Trainer

- Similar to the friend and workout groups features the Trainers have some fragments within themselves they can access
    - Search: Users can scroll through existing trainers that are signed up on the site. They can view the description of said trainer in order to decide whether or not they are interested in what the trainer is most experienced in. If so, a User can choose to follow the trainer and get a more in depth view of what workout templates they provide for users to view and use.
    - Trainers: From this page, the users can see details about the trainers description, training specialities, and the kinds of workouts they offer.
    - Following: Only viewable to the users, they can see and track every trainer that they have an interest in following. this tab will essentially show the users who they currently follow. and can view extra details about a specific trainer, or unfollow them if they choose.

User account

- The users have some functionality within themself that they can edit for more specific usages
    - Account Editing: From this the user should only have the necessary information about them pre-filled upon signup. They can choose whether or not to fill out the other fields, but specific information such as weight, height, full name, and phone number can be added to an account.
    - Password Changes: A user can choose to edit ones password in whenever they please. This tab confirms that the old password is used as confirmation, as well as making sure the password is confirmed and the password strength is met.
    - Account Deletion: Is a user chooses, they have the ability to remove themselves and their data from the app, which is accessible from this tab. There is an asked confirmation if the user is sure in the case this button is clicked by accident.
    - Wight/Musscle group classification: If a user chooses to fill in their weight, the database automatically assigns a user to a weight class which is used to separate users of different weight classes within our tournaments feature, to give the most fair experience when choosing to compete with one another. If no weight is filled in, the user is automatically assigned to the lowest weight class.

Backend (implemented)

Communication

The backend uses HTTP requests and WebSocket for database interaction:

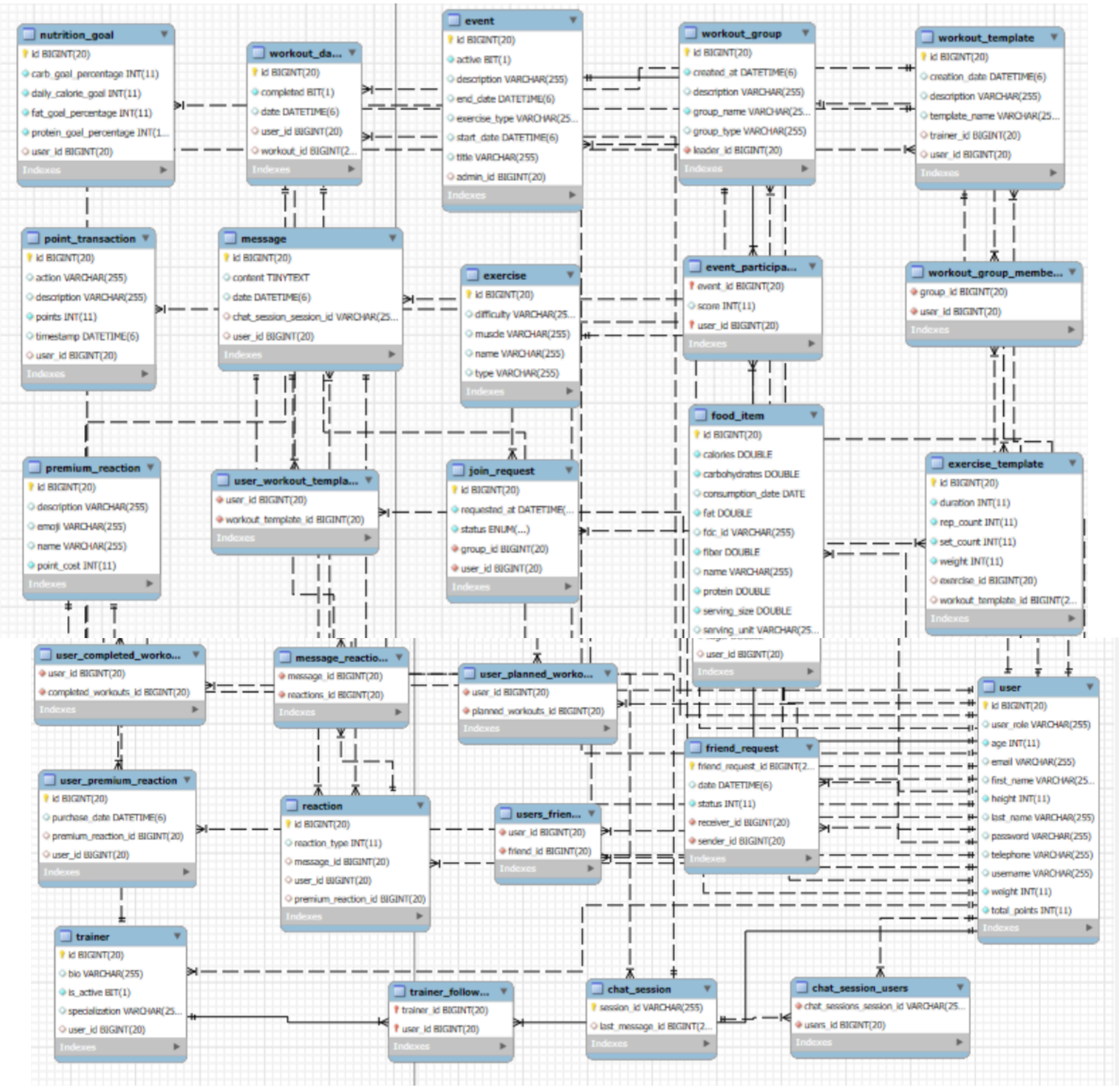
- **HTTP Requests**: POST (create), GET (retrieve), PUT (update), DELETE (remove)
- **WebSocket**: Real-time messaging for group chats and status updates

Controllers

- **Users**: Authentication, profile management
- **Calendar/Events**: Workout scheduling and notifications
- **Chat**: Individual and group messaging
- **Exercise Templates**: Predefined workout routines
- **Friend Requests**: Connection management
- **Nutrition**: Dietary tracking
- **Points**: Achievement tracking
- **Workout Groups**: Group management
- **Trainer**: Training assignments

Database

SQL database stores user data, exercise info, social connections, nutrition data, and achievements as shown in the database diagram. Tables feature appropriate foreign key relationships to maintain data integrity and support the application's social and fitness tracking features.


nutrition_goal
id BIGINT(20)
carb_goal_percentage INT(11)
daily_calorie_goal INT(11)
fat_goal_percentage INT(11)
protein_goal_percentage INT(1...
user_id BIGINT(20)
Indexes
workout_da...
id BIGINT(20)
completed BIT(1)
date DATETIME(6)
user_id BIGINT(20)
workout_id BIGINT(2...
Indexes
event
id BIGINT(20)
active BIT(1)
description VARCHAR(255)
end_date DATETIME(6)
exercise_type VARCHAR(25...
start_date DATETIME(6)
title VARCHAR(255)
admin_id BIGINT(20)
Indexes
workout_group
id BIGINT(20)
created_at DATETIME(6)
description VARCHAR(255)
group_name VARCHAR(25...
group_type VARCHAR(255)
leader_id BIGINT(20)
Indexes
workout_template
id BIGINT(20)
creation_date DATETIME(6)
description VARCHAR(255)
template_name VARCHAR(25...
trainer_id BIGINT(20)
user_id BIGINT(20)
Indexes
point_transaction
id BIGINT(20)
action VARCHAR(255)
description VARCHAR(255)
points INT(11)
timestamp DATETIME(6)
user_id BIGINT(20)
Indexes
message
id BIGINT(20)
content TINYTEXT
date DATETIME(6)
chat_session_session_id VARCHAR(25...
user_id BIGINT(20)
Indexes
exercise
id BIGINT(20)
difficulty VARCHAR(25...
muscle VARCHAR(255)
name VARCHAR(255)
type VARCHAR(255)
Indexes
event_participa...
event_id BIGINT(20)
score INT(11)
user_id BIGINT(20)
Indexes
workout_group_membe...
group_id BIGINT(20)
user_id BIGINT(20)
Indexes
food_item
id BIGINT(20)
calories DOUBLE
carbohydrates DOUBLE
consumption_date DATE
fat DOUBLE
fdc_id VARCHAR(255)
fiber DOUBLE
name VARCHAR(255)
protein DOUBLE
serving_size DOUBLE
serving_unit VARCHAR(25...
user_id BIGINT(20)
Indexes
premium_reaction
id BIGINT(20)
description VARCHAR(255)
emoji VARCHAR(255)
name VARCHAR(255)
point_cost INT(11)
Indexes
user_workout_templa...
user_id BIGINT(20)
workout_template_id BIGINT(20)
Indexes
join_request
id BIGINT(20)
requested_at DATETIME(...
status ENUM(...)
group_id BIGINT(20)
user_id BIGINT(20)
Indexes
exercise_template
id BIGINT(20)
duration INT(11)
rep_count INT(11)
set_count INT(11)
weight INT(11)
exercise_id BIGINT(20)
workout_template_id BIGINT(2...
Indexes
user_completed_worko...
user_id BIGINT(20)
completed_workouts_id BIGINT(20)
Indexes
message_reactio...
message_id BIGINT(20)
reactions_id BIGINT(20)
Indexes
user_planned_worko...
user_id BIGINT(20)
planned_workouts_id BIGINT(20)
Indexes
user
id BIGINT(20)
user_role VARCHAR(255)
age INT(11)
email VARCHAR(255)
first_name VARCHAR(25...
height INT(11)
last_name VARCHAR(255)
password VARCHAR(255)
telephone VARCHAR(255)
username VARCHAR(255)
weight INT(11)
total_points INT(11)
Indexes
friend_request
friend_request_id BIGINT(2...
date DATETIME(6)
status INT(11)
receiver_id BIGINT(20)
sender_id BIGINT(20)
Indexes
user_premium_reaction
id BIGINT(20)
purchase_date DATETIME(6)
premium_reaction_id BIGINT(20)
user_id BIGINT(20)
Indexes
reaction
id BIGINT(20)
reaction_type INT(11)
message_id BIGINT(20)
user_id BIGINT(20)
premium_reaction_id BIGINT(20)
Indexes
users_frien...
user_id BIGINT(20)
friend_id BIGINT(20)
Indexes
trainer
id BIGINT(20)
bio VARCHAR(255)
is_active BIT(1)
specialization VARCHAR(25...
user_id BIGINT(20)
Indexes
trainer_follow...
trainer_id BIGINT(20)
user_id BIGINT(20)
Indexes
chat_session
session_id VARCHAR(255)
last_message_id BIGINT(2...
Indexes
chat_session_users
chat_sessions_session_id VARCHAR(25...
users_id BIGINT(20)
Indexes